

國立雲林科技大學



電子工程系 碩士班

樹莓派之遠端控制 LED 燈

指導教授：許明華 教授

王斯弘 講師

電子碩二 M10713203 李政聲

電子碩一 M10713245 江閔詳

工管碩一 M10721212 賴淑玲

目錄

第一章	緒論	3
1.1	研究背景與動機.....	3
1.2	研究目的.....	3
第二章	研究方法	4
2.1	實驗材料與系統.....	4
2.2	實驗流程.....	4
第三章	實驗結果	11
第四章	學習心得	16
第五章	參考資料	17

第一章 緒論

1.1 研究背景與動機

隨著物聯網的快速發展，使物聯網應用越來越多元，例如智慧工廠、環境偵測以及智能家庭等。樹莓派是一款基於 Linux 系統的單機版電腦，它的價位低、體積小、擴充性佳、支援多種語言且應用非常廣泛，許多人運用樹莓派設計出各種有趣的東西，例如：家電遙控、雲端相機、自動灑水器等，由上述可知樹莓派是一個可以無限發揮的平台，非常適合進行各式實驗，因此本研究希望應用樹莓派進行一實際操作。

1.2 研究目的

本實驗利用 Android Studio 作為用戶端的操作系統，結合 Google 旗下的 Firebase 以便遠端操作，透過觸控與語音方式操作造型燈的開關與顏色變化。藉由本實驗學習到如何將樹莓派、GPIO 控制、Firebase 與 Android Studio 結合以達到遠端操作。

第二章 研究方法

本章主要對於本實驗所使用的材料和系統與實驗流程做介紹。

2.1 實驗材料與系統

此實驗使用的材料為樹莓派 3、RGB LED 燈、杜邦線、麵包板、SD 卡、Android 系統之手機、5V/2A USB 電源轉換器、HDMI 線、USB 鍵盤、滑鼠與螢幕。此實驗使用的系統為 Raspbian Stretch Lite、Win32 Disk Imager、Firebase、Android Studio。主要語言為 Python 與 Java。

樹莓派是一款基於 Linux 系統的單機版電腦實驗材料，它的價位低、體積小、擴充性佳、支援多種語言且應用非常廣泛，樹莓派上包含四個 USB 接口、一個 HDMI、一個 Ethernet、一個 Micro USB 與一組 40 個接腳的 GPIO，可以提供使用者經由程式來控制。

Firebase 為 Google 所提供可同時支援多種開發環境與工具的雲端平台，所提供的功能也眾多，包括 Realtime Database 即時資料庫、Cloud Storage 雲端儲存與 Analytics 數據分析，而使用 Firebase 作為 APP 後端平台的人數不斷成長。

2.2 實驗流程

本實驗分為三部分進行，第一部分為環境與設備之架設，第二部分為 LED 燈控制與雲端連結，第三部分為開發用戶端的操作系統，下圖為實驗流程圖。

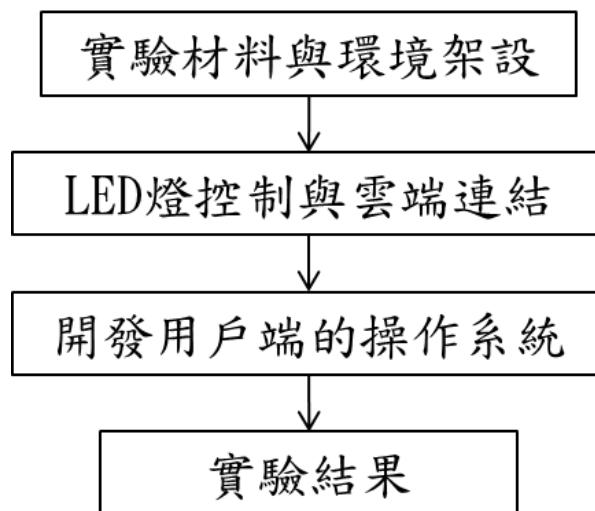


圖 2-1 流程圖

第一部分我們使用 Raspbian 這套作業系統，首先下載 Stretch Lite 映像檔，透過 Win32 Disk Imager 程式將該映像檔寫入至 SD 卡中，就成功將該系統安裝至樹莓派上，將燒錄好的 SD 卡、鍵盤、滑鼠、螢幕與電源接上，即可啟動電源，啟用電源後，可以使用 TERMINAL 進行相關的察看與設定。

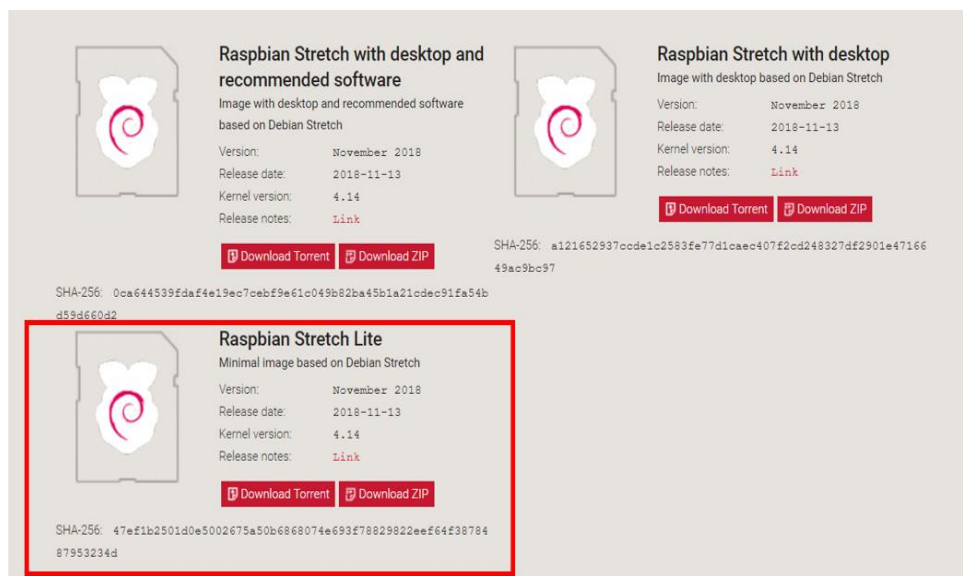


圖 2-2 Raspbian Stretch Lite 映像檔下載

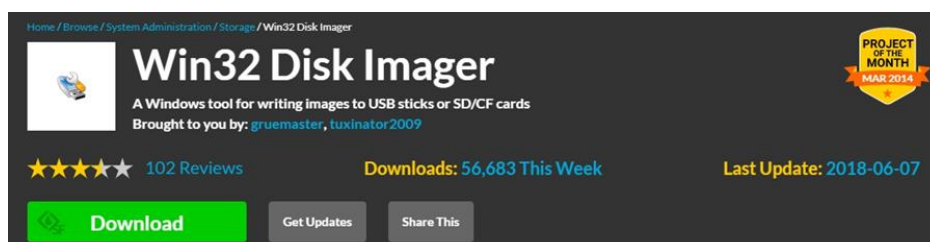


圖 2-3 Win32 Disk Imager 程式寫入映像檔

第二部分我們使用樹莓派上之 GPIO 來控制 LED 燈，並連結至 Firebase，首先在 Firebase 建立一個專案，並更改讀寫權限，接著新增所需的資料欄與變數，以便與樹莓派做連結，接著在樹莓派上寫一個 Python 程式並依照 Firebase 上之變數來控制開關。以下圖 2-4、圖 2-5 與圖 2-6 為新增 Firebase 資料欄與連結 Firebase 並依照變數來控制開關與閃爍方式之控制的程式碼之註解。

1. 新增 Firebase 的 led1 與 led2 資料欄。
2. import 套件。
3. 新增一個變數存放 Firebase 的連結。
4. 設定 led 變數，led1 為 17、22、27 腳位，led2 為 14、15、18 腳位。
5. 初始化，在開始執行前先將 led 燈全部關閉。
6. 新增一個變數，用 get 方式去讀取的網址，並轉成 json 格式，再取得 Firebase 欄位 led1 與 led2 下的資料。
7. 取得 led1 的 17、22、27 與 led2 的 14、15、18 腳位的紅綠藍狀態。
8. 使用 Flashing 來增加閃爍功能，並用 if else 判斷式來判斷閃爍方式。

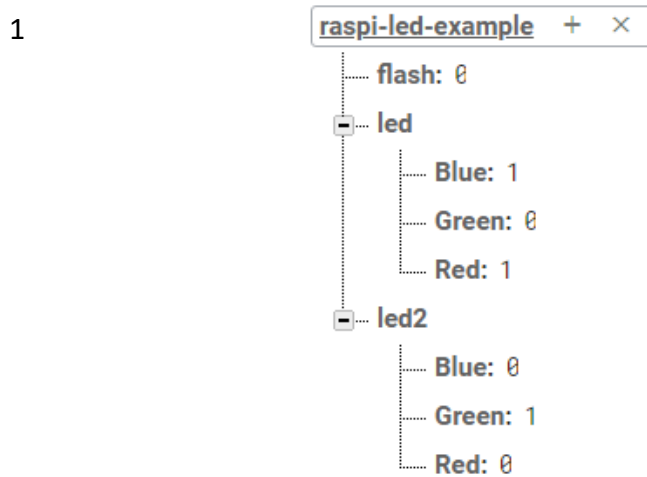


圖 2-4 新增 Firebase led1 與 led2 資料欄

2

```

#coding=utf-8
from gpiozero import LED
from requests import get
import json
import time

```

3

```

#firebase 資料庫 url
Firebase_URL = 'https://raspi-led-example.firebaseio.com/.json'

```

4

```

led1_17 = LED(17) #LED1_Green
led1_22 = LED(22) #LED1_Blue
led1_27 = LED(27) #LED1_Red

led2_14 = LED(14) #LED2_Green
led2_15 = LED(15) #LED2_Blue
led2_18 = LED(18) #LED2_Red

```

5

```

#初始化 LED 燈
led1_17.off()
led1_22.off()
led1_27.off()
led2_14.off()
led2_15.off()
led2_18.off()

```

6

```

while True:
    #讀取 firebase 上資料庫的資料
    Led1_Data = get(Firebase_URL).json()['led']
    Led2_Data = get(Firebase_URL).json()['led2']
    flash = get(Firebase_URL).json()['flash'];

```

7

```

#燈號顯示
led1_17.value = Led1_Data['Green']
led1_22.value = Led1_Data['Blue']
led1_27.value = Led1_Data['Red']
led2_14.value = Led2_Data['Green']
led2_15.value = Led2_Data['Blue']
led2_18.value = Led2_Data['Red']

```

圖 2-5 連結 Firebase 並依照變數來控制開關的程式碼

```

#閃爍功能
if flash == 1:
    count=0
    for count in range(0,8):
        if count%2==0:
            led1_17.value=0
            led2_14.value=0
        else:
            led1_17.value=1
            led2_14.value=1
        if count<4:
            led1_22.value=0
            led2_15.value=0
        else:
            led1_22.value=1
            led2_15.value=1
        if count==0 or count==1 or count==4 or count==5:
            led1_27.value=0
            led2_18.value=0
        else:
            led1_27.value=1
            led2_18.value=1
    time.sleep(.300)
if flash == 0:
    led1_17.value=0
    led2_14.value=0
    led1_22.value=0
    led2_15.value=0
    led1_27.value=0
    led2_18.value=0

```

圖 2-6 使用 if else 判斷式來判斷閃爍方式的程式碼

第三部分我們使用 Android Studio 來開發此環境，首先安裝 Android Studio，再建立一新專案再設定手機模擬器並與 Firebase 做連結，接著將所需之元件配置到手機介面上，介面設計完成後撰寫 java 程式，即可透過手機的觸控與語音來控制 LED 燈的開關與顏色變化。以下圖 2-7 至圖 2-10 程式碼之註解。

```

package com.example.course_iot_rgbled;

import android.content.Intent;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

```

圖 2-7 引用函式庫

```

public class MainActivity extends AppCompatActivity {

    private Spinner sprLED1, sprLED2;
    private Button bSpeech, bFlash;
    private int isFlash = 0;

    FirebaseDatabase fireDB = FirebaseDatabase.getInstance();
    DatabaseReference led1_R, led1_G, led1_B, led2_R, led2_G, led2_B, flash;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ImplementsIni();
        FireDBSetup();

        // Spinner declaration
        ArrayAdapter adapter = ArrayAdapter.createFromResource(
            this,
            R.array.Colors,
            R.layout.support_simple_spinner_dropdown_item);

        sprLED1.setAdapter(adapter);
        sprLED2.setAdapter(adapter);

        // LED_1 Spinner onSelected Listener
        sprLED1.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                switch (position){
                    case 0:
                        led1_R.setValue(0);
                        led1_G.setValue(0);
                        led1_B.setValue(0);
                        break;
                    case 1:
                        led1_R.setValue(1);
                        led1_G.setValue(0);
                        led1_B.setValue(0);
                        break;
                    case 2:
                        led1_R.setValue(0);
                        led1_G.setValue(1);
                        led1_B.setValue(0);
                        break;
                    case 3:
                        led1_R.setValue(0);
                        led1_G.setValue(0);
                        led1_B.setValue(1);
                        break;
                }
            }
        });
    }
}

```

圖 2-8 初始設定、下拉式選單之功能設定 (重複部分省略)


```

// Speech Button onClick Listener
public void bSpeechClick(View view) {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Start Speech");
    startActivityForResult(intent, 1);
}

// Flash Button onClick Listener
public void onFlashClick(View view){
    isFlash = (isFlash == 0)?1:0;
    flash.setValue(isFlash);
}

// Speech activity
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    final String off = "關";
    final String on = "開";

    if (requestCode == 1 && resultCode == RESULT_OK){
        String firstMatch = data != null
            ? data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS).get(0)
            : null;

        flash.setValue(0);
        isFlash = 0;

        if(firstMatch.contains(on)){
            led1_R.setValue(1);
            led1_G.setValue(1);
            led1_B.setValue(1);
            led2_R.setValue(1);
            led2_G.setValue(1);
            led2_B.setValue(1);
        }else if(firstMatch.contains(off)){
            led1_R.setValue(0);
            led1_G.setValue(0);
            led1_B.setValue(0);
            led2_R.setValue(0);
            led2_G.setValue(0);
            led2_B.setValue(0);
        }
    }
}

```

※注意：這邊我們只做一個閃爍用的 button，原因是我們認為一個 button 就能完成開關閃爍的功能（即按一下打開，再按一下關閉），而我們的實驗結果也同樣完成了作業所要求的功能！

圖 2-9 錄音功能及閃爍按鈕

```

// Implements initialization
private void ImplementsIni () {
    sprLED1 = findViewById(R.id.sprLED1);
    sprLED2 = findViewById(R.id.sprLED2);
    bSpeech = findViewById(R.id.bSpeech);
    bFlash = findViewById(R.id.bFlash);
}

// FireBase Setup
private void FireDBSetup () {
    led1_R = fireDB.getReference().child("led").child("Red");
    led1_G = fireDB.getReference().child("led").child("Green");
    led1_B = fireDB.getReference().child("led").child("Blue");
    led2_R = fireDB.getReference().child("led2").child("Red");
    led2_G = fireDB.getReference().child("led2").child("Green");
    led2_B = fireDB.getReference().child("led2").child("Blue");
    flash = fireDB.getReference().child("flash");
}

```

圖 2-10 物件初始化、FireBase 設定

第三章 實驗結果

在完成上述所有實驗流程後，以下為實驗結果展示。

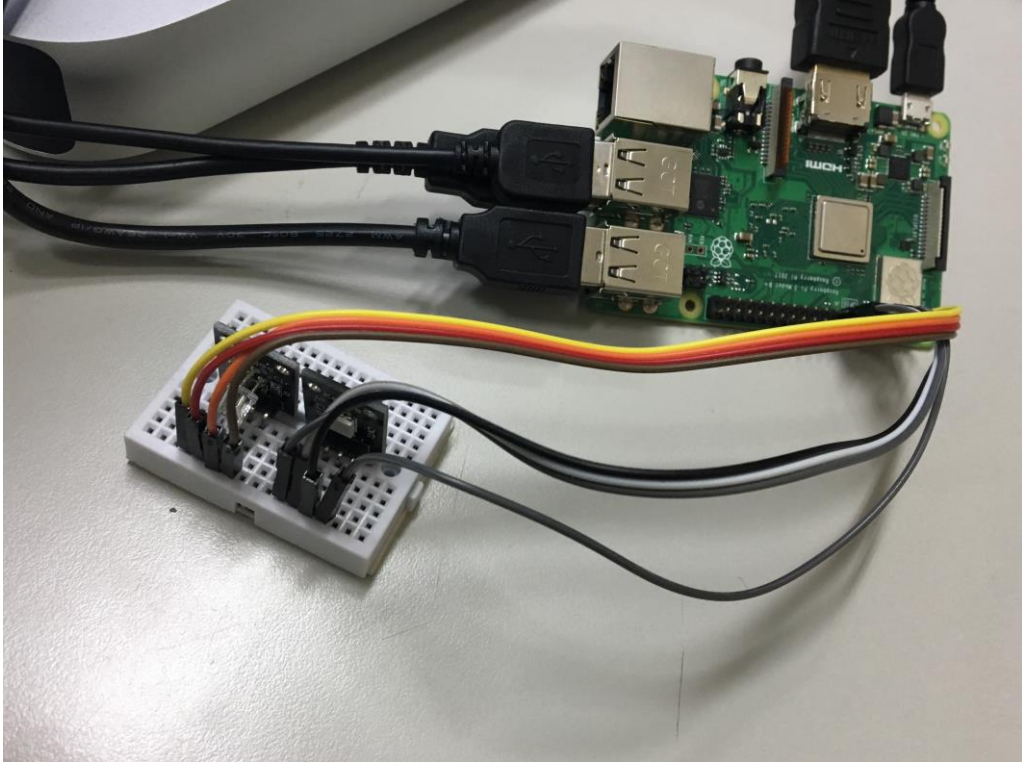


圖 3-1 硬體

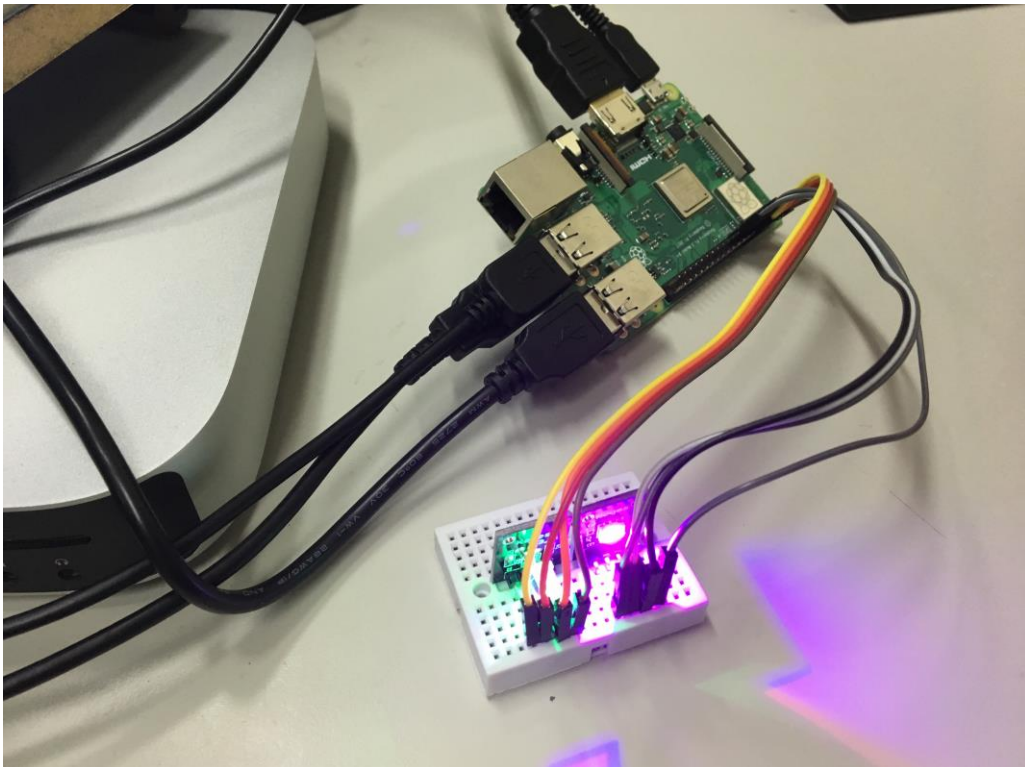


圖 3-2 閃爍結果



圖 3-3 手機介面



圖 3-4 燈號選擇 1



圖 3-5 燈號選擇 2

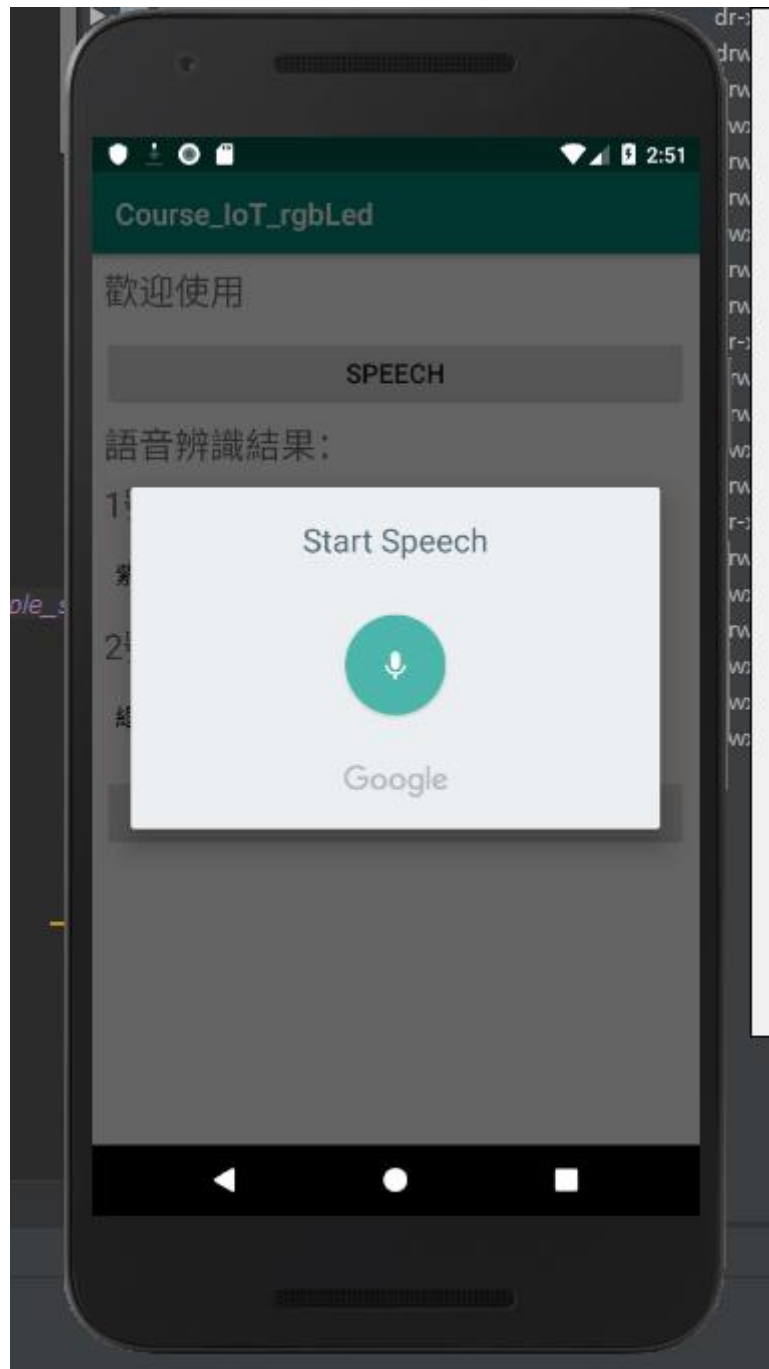


圖 3-6 錄音功能

第四章 學習心得

李政聲：

本實驗利用樹莓派結合現成的資料庫 firebase 與 Android App 來完成，雖完以前有寫過 Android，但還是學習到了一些與資料庫串接與官方套件(錄音)的應用，另外，我們也學習到了 python 的基本語法，真的收穫豐富，老師上課內容充實且清楚，很高興有機會能修到這門課，且由於往後的課程是有關於 AIoT，這與我的碩士論文也有些關聯，所以接下來的實驗我會更努力學習。

江閔詳：

在這門課中，真的可以接觸及學習到很多東西，要使用多方程式及裝置完成作業，真的是受益良多。在這次課程，接觸到樹莓派、Python、Firebase 及 android，這是我第一次接觸到樹莓派，沒想到這小小板子可以有很多的應用，用 Python 連線至 Firebase 讀取資料，然後控制 LED，在使用第三方裝置手機控制 Firebase 資料，這樣子就完成了聯網，利用這方法，就可以將這簡單的控制應用在生活中，例如電源控制、環境檢測，原來聯網就這麼簡單。謝謝老師的教導，讓我學習到更多得方法。

賴淑玲：

樹莓派的操作容易且攜帶方便，它提供我們一個很大的空間讓我們進行設計，且可以輕易的對硬體進行擴充並連結網路，這次實驗完整操作整個流程，包括從環境架設、設備準備、程式撰寫、資料庫連結與 APP 設計等，因此透過本實驗學習到樹莓派、GPIO 控制、Firebase 與 Android Studio 的結合以達到遠端操作，是一個非常有趣且吸收到非常多知識的實驗。

第五章 參考資料

1. 樹莓派- 維基百科:
<https://zh.wikipedia.org/wiki/%E6%A0%91%E8%8E%93%E6%B4%BE>
2. Raspberry Pi 台灣樹莓派:
<https://www.raspberrypi.com.tw/>
3. Raspberry Pi -Teach, Learn, and Make with Raspberry Pi:
<https://www.raspberrypi.org/>
4. 葉難: Raspberry Pi 安裝中文環境:
<http://yehnan.blogspot.tw/2012/08/raspberry-pi.html>
5. GPIO: MODELS A+, B+, RASPBERRY PI 2 B AND RASPBERRY PI 3 B:
<https://www.raspberrypi.org/documentation/usage/gpio-plus-andraspi2/README.md>
6. 透過 Firebase 與 Raspberry Pi 製作簡單的物聯網 iOS 程式:
<https://www.appcoda.com.tw/swift-raspberry-pi/>
7. Firebase:
<https://firebase.google.com/docs/android/setup>
8. 使用 Android Studio 2.2.2 開發 Firebase - Database 篇:
https://litotom.com/2016/11/19/as_222_firebase_db/
9. gpiozero:
<https://gpiozero.readthedocs.io/en/stable/>
10. RASPBERRY PI 3 MODEL B 系統安裝指南:
<http://hophd.com/raspberry-pi-3-model-b-installation/>
11. 啟動 Android 的語音識別(Speech Recognition)功能:
<http://androidbiancheng.blogspot.tw/2011/05/android-speech-recognition.html>
12. android-studio 安裝教學-從 0 到 1/:
<http://learnexp.tw/%E3%80%90android%E3%80%91androidstudio%E5%AE%89%E8%A3%9D%E6%95%99%E5%AD%B8-%E5%BE%9E0%E5%88%B01/>